

# Documentación de los programas, tarea 1 elo329.

Pascal Ernesto Sigel Olivares

abril-2014

# 1 Introducción:

En el presente documento se explicará cada uno de los programas los cuales están cada uno en su carpeta personal llamada etapa(n) donde n es un número. El proyecto está separado en varias etapas, desde la número 1 hasta la número 5 y cada etapa agrega alguna característica a la anterior.

El proyecto en sí es un simulador de interacciones físicas como, por ejemplo, el choque de dos bolas, una bola junto con un resorte o un bloque con rose con bolas y resortes en cualquier disposición deseada.

En la próxima sección se mostrarán algunos detalles importantes de el programa etapa1, que consta de dos bolas una en reposo y la otra en movimiento que chocan con choque elástico.

## 2 Etapa 1

El programa etapa 1 es la primera versión de este simulador el cual simula la interacción de dos bolas que chocan. En este programa se crea la primera versión del simulador con las clases PhysicsLab, MyWorld, PhysicsElement y Ball. Se hará una descripción de cada una de estas clases.

### 2.1 Clase PhysicsLab:

La clase PhysicsLab es la clase main del proyecto, esta clase manipula el objeto de tipo MyWorld creándolo y dándole los elementos que interaccionarán en este “mundo” simulado.

### 2.2 Clase MyWorld:

Esta clase es la que maneja la interacción entre los elementos del mundo simulado, el método principal de esta clase es el método **simulate** el cual comienza la simulación. En la función **simulate** se requiere que cada elemento físico (PhysicsElement) sea capaz de calcular su próximo estado.

### 2.3 PhysicsElement:

Los elementos físicos son una clase abstracta de la cual nacen los elementos como bolas, resortes, bloques y cualquier otro elemento físico.

### 2.4 Ball:

Esta clase se extiende (desciende) de la clase PhysicsElement y representa una bola la cual tiene masa, velocidad y posición. En esta clase se calcula el próximo estado.

### 2.5 Experimento y resultados:

Lease el archivo README para ejecutar el programa.

El caso ha simular puede ser representado con la siguiente figura:

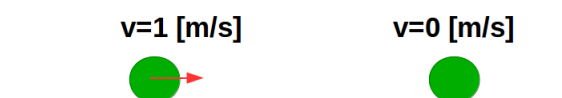


Figure 1: Simulación realizada en etapa 1, en la simulación la posición inicial de la bola con velocidad es  $x=1$ [mts] y la otra es en  $x=2.56$ [mts] ambos con radio  $0.1$  [mts].

Se ha simulado la situación descrita en la figura 1, o sea el choque elástico entre dos bolas, con la simulación se obtuvieron el siguiente gráfico.

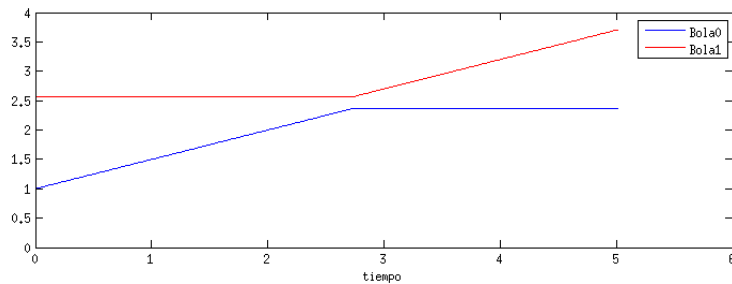


Figure 2: Resultados simulación etapa 1, Bola 0: bola inicialmente en movimiento y Bola 1: bola inicialmente en reposo.

Como se observa en el gráfico de la figura 4 cuando la bola 0 colisiona con la bola 1 (a 0.2 mts de distancia dado sus radios) la bola 1 obtiene todo el movimiento como se esperaría de un choque elástico de dos masas iguales y una en reposo.

## 2.6 Problemas que ocurrieron:

### 2.6.1 Problema con formatos:

Por defecto Java usa el formato de numeros flotantes con coma “,” pero software como matlab usan el formato flotante con punto “.” Por lo que en vez de usar `System.out.print()` se usó `System.out.format(Locale.US, ...)`.

### 2.6.2 Elección de delta t:

Si el delta t es muy grande entonces el computo de colisión no alcanza a ocurrir cuando una de las bolas ya atravezó a la otra, por ejemplo si la velocidad de la bola 0 fueran 100 [mts/s] y el delta t es 1[s] entonces en el siguiente cómputo la bola 0 a avanzado 100 mts y probablemente atravezó a la bola 1 sin la ocurrencia de colisión.

Con esto se da por terminada la primera etapa y se sigue con la siguiente etapa en la cual se agrega el elemento físico resorte.

## 3 Etapa 2

Es la segunda etapa del programa, en esta etapa se agrega los elementos físicos tipo resorte (Spring) con la clase Spring.java.

### 3.1 Clase Spring:

La clase spring extiende de la clase PhysicsElement y representa un resorte, la implementación del resorte consta de asociar cada extremo a una bola y sabiendo la posición de las bolas de cada uno de sus extremos se puede calcular la fuerza que este resorte efectúa ha cada uno de sus extremos.

### 3.2 Experimento y resultados:

Lease el archivo README para ejecutar el programa.

El caso ha simular puede ser representado con la siguiente figura:

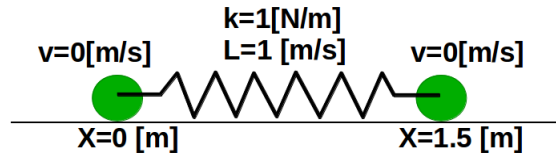


Figure 3: Simulación realizada en etapa 2, en la simulación ambas bolas comienzan en reposo una en  $x=0$  [mts] y la otra es en  $x=1.5$  [mts] ambos con radio 0.1 [mts], el resorte efectúa fuerzas según la ley de hooke sobre ambas bolas con stiffness = 1 y largo natural 1.

Se ha simulado la situación descrita en la figura 3. Con la simulación se obtuvieron el siguiente gráfico.

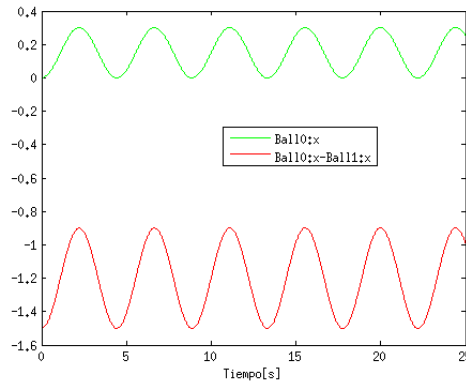


Figure 4: Resultados simulación etapa 2, en el gráfico se muestran dos diferentes datos en uno la posición de la bola que está inicialmente situada en 0 y la otra es la diferencia entre las bolas.

Se observa un comportamiento deseado: ambas bolas oscilan y además su distancia también oscila demostrando que el resorte se contrae y se acercan y se expande como se esperaría de la situación ideal presentada.

Además de esta situación se simuló otra situación más con bolas y resortes: dos bolas unidas con un resorte, una de las bolas tiene velocidad inicial por lo que el momentum del sistema tiene dirección (así este sistema se ira moviendo al mismo tiempo en que se contrae y se expande) y además hay una tercera bola que inicialmente está en reposo y que posteriormente es colisionada por este sistema. Los resultados de este otro experimento son los siguientes.

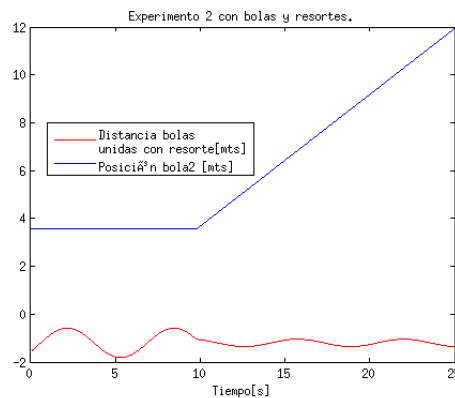


Figure 5: Experimento alternativo etapa2: dos bolas unidas con resorte tienen momentum inicial (dado que una de las bolas tiene velocidad inicial) colisionan una tercera bola (bola2) que toma parte de la energía del sistema y sale con velocidad constante, luego el sistema al haber perdido momentum oscila con menor amplitud.

El resultado es bueno dado que se observa como el sistema entrega parte de su momentum (y su energía) a la bola2 (tercera bola que estaba en reposo).